

(19)



JAPANESE PATENT OFFICE

PATENT ABSTRACTS OF JAPAN

(11) Publication number: 06187145 A

(43) Date of publication of application: 08.07.94

(51) Int. Cl

G06F 9/06

G06F 9/06

(21) Application number: 04338696

(22) Date of filing: 18.12.92

(71) Applicant: TOSHIBA CORP

(72) Inventor: HIRAYAMA MASAYUKI  
OGASAWARA HIDETO

(54) SOFTWARE QUALITY EVALUATION SYSTEM

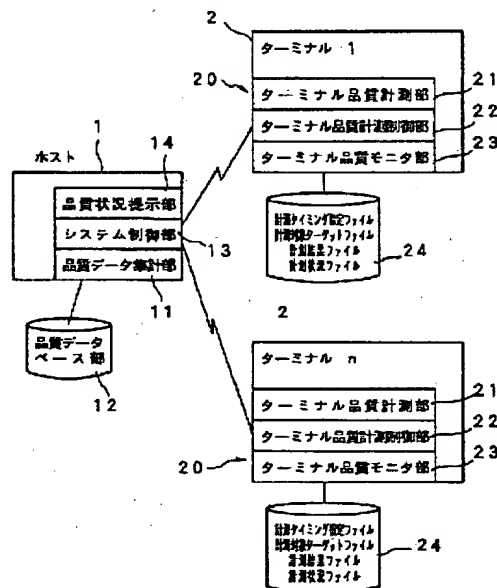
(57) Abstract:

PURPOSE: To easily evaluate the quality of an entire software even in the case of developing the software under decentralized environments by automatically gathering quality data to a manager side successively.

CONSTITUTION: The quality evaluation of the software distributed and developed respectively in plural terminal equipments 2 is executed through a host device 1 connected to the respective terminal equipments 2. First, measurement timing specifying files are transferred from the host device 1 to the respective terminal equipments 2. The respective terminal equipments 2 are provided with a quality condition monitor 20 for grasping quality information relating to respective distributed nodes, a result measured corresponding to the specification of the host device 1 is tentatively kept in the storage part 24 of the respective terminal equipments 2 and a measurement completion flag is raised. At the host device 1, after confirming the flags of the entire relating terminal equipments 2, measured result files tentatively kept in the respective terminal equipments 2 are transferred

through a network to the host device 1.

COPYRIGHT: (C)1994,JPO&Japio



(19)日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11)特許出願公開番号

特開平6-187145

(43)公開日 平成6年(1994)7月8日

(51)Int.Cl.<sup>5</sup>

G 0 6 F 9/06

識別記号

4 4 0 V 9367-5B

4 3 0 E 9367-5B

庁内整理番号

F I

技術表示箇所

審査請求 未請求 請求項の数1(全 9 頁)

(21)出願番号

特願平4-338696

(22)出願日

平成4年(1992)12月18日

(71)出願人 000003078

株式会社東芝

神奈川県川崎市幸区堀川町72番地

(72)発明者 平山 雅之

神奈川県川崎市幸区柳町70番地 株式会社

東芝柳町工場内

(72)発明者 小笠原 秀人

神奈川県川崎市幸区柳町70番地 株式会社

東芝柳町工場内

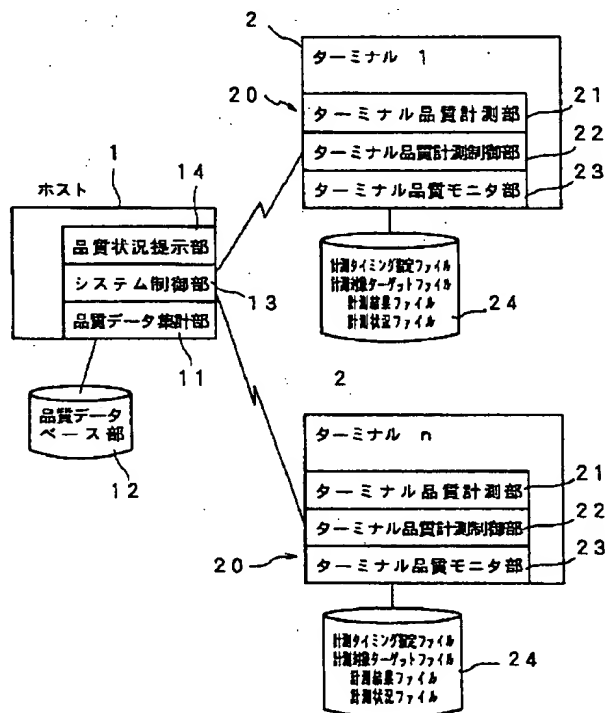
(74)代理人 弁理士 須山 佐一

(54)【発明の名称】 ソフトウェア品質評価システム

(57)【要約】 (修正有)

【目的】 分散環境下でソフトウェア開発を行う場合であっても、ソフトウェア全体の品質評価を容易に行うことのできるソフトウェア品質評価システムを提供する。

【構成】 ホスト装置1には、ターミナル装置から品質データを集計する品質データ集計部11、各品質データを保管する品質データ収容手段12、本システム全体の動作制御を司るシステム制御部13、及び全体の品質情報を提供する品質状況提示部14が設けられている。ターミナル装置2には、開発した成果物の品質計測を行う品質状況モニタ20が設けられている。



## 1

## 【特許請求の範囲】

【請求項1】 複数のターミナル装置でそれぞれ分散されて開発されるソフトウェアの品質評価を、前記各ターミナル装置と接続されたホスト装置を介して実行するソフトウェア品質評価システムであって、前記各ターミナル装置は、これらのターミナル装置で開発された成果物の品質計測を個別に行う品質状況モニタ手段を具備し、前記ホスト装置は、前記各品質状況モニタ手段に品質計測タイミングを指定して、これらの品質状況モニタ手段に所定の品質計測タイミングにて品質計測を実行させ、この品質計測の結果である品質データを収集するシステム制御手段と、前記システム制御手段によって収集された品質データを集計する品質データ集計手段と、前記システム制御手段によって収集された品質データおよび前記品質データ集計手段によって集計された集計結果を収容する品質データ収容手段と、前記品質データ集計手段によって集計された集計結果を提示する品質状況提示手段とを具備したことを特徴とするソフトウェア品質評価システム。

## 【発明の詳細な説明】

## 【0001】

【産業上の利用分野】 本発明は、複数人により、分散環境下で開発されたソフトウェアの品質評価を行うソフトウェア品質評価システムに関する。

## 【0002】

【従来の技術】 近年のソフトウェアの開発では、複数人による分散環境下での開発が増加している。このような開発形態をとった場合、開発途上での品質管理を行おうとすると、管理者は、個々の開発担当者に品質状況を問い合わせ、データ収集等を行う必要があり、多くの手間がかかり、多大なオーバーヘッドが生じ、場合によっては開発作業そのものにまで影響を及ぼす場合がある。

【0003】 また、従来提案されているソフトウェア品質評価システムでは、このような分散環境を対象としているものは少なく、また、品質管理データを人間系で入力し、その管理を支援する機能を主とするものである。

## 【0004】

【発明が解決しようとする課題】 上述したように、従来においては、分散環境において開発されているソフトウェアの品質を、開発途上で実行することが困難であり、このような品質評価を容易に行うことのできるソフトウェア品質評価装置の開発が望まれていた。

【0005】 本発明は、かかる従来の事情に対処してなされたもので、分散環境下でソフトウェア開発を行う場合であっても、ソフトウェア全体の品質評価を容易に行うことのできるソフトウェア品質評価システムを提供しようとするものである。

## 【0006】

## 2

【課題を解決するための手段】 すなわち、本発明のソフトウェア品質評価システムは、複数のターミナル装置でそれぞれ分散されて開発されるソフトウェアの品質評価を、前記各ターミナル装置と接続されたホスト装置を介して実行するソフトウェア品質評価システムであって、前記各ターミナル装置は、これらのターミナル装置で開発された成果物の品質計測を個別に行う品質状況モニタ手段を具備し、前記ホスト装置は、前記各品質状況モニタ手段に品質計測タイミングを指定して、これらの品質状況モニタ手段に所定の品質計測タイミングにて品質計測を実行させ、この品質計測の結果である品質データを収集するシステム制御手段と、前記システム制御手段によって収集された品質データを集計する品質データ集計手段と、前記システム制御手段によって収集された品質データおよび前記品質データ集計手段によって集計された集計結果を収容する品質データ収容手段と、前記品質データ集計手段によって集計された集計結果を提示する品質状況提示手段とを具備したことを特徴とする。

## 【0007】

【作用】 上記構成の本発明のソフトウェア品質評価システムでは、分散環境下でのソフトウェア開発を行う場合であっても、逐次品質データが管理者サイドに自動的に収集されるため、従来のようなデータ収集に要するオーバーヘッドが軽減される。 また、従来は、品質確認が必要と判断した時点で、管理者が品質状況の確認、品質データの収集を実施していたため、対策が後手にまわる場合が多く、品質評価結果がソフトウェア開発に有効に反映されていないケースが多かった。これに対して、本発明のソフトウェア品質評価システムでは、管理者が予めスケジューリングしておくことにより、自動的にその時点での品質データが計測、収集されるため、分散環境下での開発であっても、開発途中で管理者が容易にその品質状況（全体としてのソフトウェアの複雑さ、理解し易さ等）を確認することが可能となり、問題点が発生する以前の早期に対策をたてることが可能となる。

【0008】 なお、本発明のソフトウェア品質評価装置では、ソフトウェアの品質評価の観点として、設計・作成工程等のソフトウェア開発の途中で作成される成果物を対象とし、これらの理解し易さ、保守し易さ、複雑さ等の観点を中心として評価を行う。

【0009】 これらの中間段階での成果物が理解しにくい、あるいは、複雑な構造になっている場合には、後工程での誤解、理解不足等によってバグが混入する可能性が高い。また、保守しにくい成果物では、試験工程で発見された問題への対処がしにくいという問題、さらに、リリース後の保守・変更がやりにくい等の問題が発生し、結果的に製品ソフトウェアとしては、品質面からの問題を持つことになる。本発明のソフトウェア品質評価装置では、開発途中の成果物をその作成段階で評価することによって、未然に品質の問題点を除去することがで

きる。

【0010】ソフトウェアの品質指標としては、製品ソフトウェアに関するバグの数等が重視されているが、本発明のソフトウェア品質評価装置で扱う開発途中での成果物の理解性、保守性、複雑さ等を早期に評価することは、結果的に、製品ソフトウェア自身のバグを未然に防ぐ等品質を向上させることにつながる。

【0011】なお、ソフトウェアの品質の観点としては、上述した成果物の理解性、保守性、複雑さ等以外にも、製品ソフトウェアのバグ数等に代表される信頼性、あるいは移植性等の様々な観点が提案されている。ソフトウェアの品質評価は、これらの全ての観点を網羅した形で計測、評価することが望ましい。

【0012】

【実施例】以下、本発明の一実施例を図面を参照して説明する。

【0013】図1は、本発明の一実施例のソフトウェア品質評価システムの構成を示すもので、同図において、1はホスト装置、2は分散して設けられ各々このホスト装置ネットワークにより接続された複数（n個）のターミナル装置である。

【0014】上記ホスト装置1には、品質データを集計する品質データ集計部11と、各品質データを保管する品質データベース12と、システム制御部13と、全体の品質情報を提供する品質状況提示部14が設けられている。

【0015】上記システム制御部13は、本システム全体の動作制御を司るものであり、以下に示すような役割を持つ。

【0016】ターミナル装置2に対する計測タイミング指定ファイルの転送。

【0017】ターミナル装置2における品質計測状況の確認。

【0018】ターミナル装置2における品質計測終了後の品質計測結果の吸い上げ。

【0019】ホスト装置1での品質データ集計部11の起動制御。

【0020】ホスト装置1での品質データ集計終了後の品質状況提示部の起動制御。

【0021】ホスト装置1での計測結果データ、集計データの品質データベースへの格納処理制御。

【0022】一方、ターミナル装置2には、分散した各ノード（個人）に関する品質情報を把握するための品質状況モニタ20が設けられている。この品質状況モニタ20は、各担当者が開発した成果物の品質計測を行うターミナル品質計測部21と、品質状況モニタ20全体の動作制御を行うターミナル品質計測制御部22と、品質計測結果をターミナル装置2のディスプレイ等に提示可能な形の品質状況シートの作成を行うターミナル品質モニタ部23とから構成されている。また、このターミナ

ル装置2には、後述する計測タイミング指定ファイル、計測対象ターゲットファイル、計測結果ファイル、計測状況ファイル等を収容する記憶部24が設けられている。

【0023】上記構成のこの実施例のソフトウェア品質評価システムにおいては、分散環境下においてターミナル装置2により各個人が開発を行った対象成果物の品質について、その状況を自動計測するが、自動計測を行う対象成果物は、図2に示すように、定型書式（フローチャート、TFF、HIPO等）に従って記述された定型書式设计書3.1ならびにC言語、FORTRAN、PL/M、ASSEMBLA等によるソースコード3.2である。

【0024】各個人は、一つのソフトウェア開発プロジェクトに関する設計／コーディング工程に入る前に、計測対象ターゲットファイルを作成し、この中に上記対象成果物が各ノード（ターミナル装置2）のどこに存在しているかを宣言する。図3に示すように、計測対象ターゲットファイルは、実際の開発を自分の使用する装置のどのディレクトリに定型書式设计書あるいはソースコードが存在するかを示したものである。図3の例では、ソースコードはYesq¥source ディレクトリに、定型書式设计書はYesq¥tffdesignにある場合を示している。

【0025】一方、ホスト装置1側においては、管理者は、予め図4に示すような計測タイミング指定ファイルを作成しておく。この計測タイミング指定ファイルは、どのようなタイミングで品質計測を実行するかを示すもので、図4に示す例では、92年の8月1日、8月15日、9月18日等に品質計測を実行することを示している。

【0026】次に、図5のフローチャートを参照して本ソフトウェア品質評価システムの動作について説明する。

【0027】まず、本システムのイニシャライズ時に、上記した計測タイミング指定ファイルが、ホスト装置1から各ターミナル装置2に転送される（101）。

【0028】この転送は、ホスト装置1と各ターミナル装置2との間のネットワークを利用し、ターミナル装置2側の本システム実行ディレクトリにファイルコピーの形で行う。なお、ホスト装置1とターミナル装置2の両方の実行環境を整備するため、本システムのイニシャライズは、ホスト装置1とターミナル装置2の双方が起動している状態で行う。

【0029】各ターミナル装置2では、ターミナル品質計測制御部22により、上述のようにして転送された計測タイミング指定ファイルを常時サーチする（201）。

【0030】そして、ターミナル装置2内のカレンダー機能による日付と計測タイミング指定ファイルの日付が一致した場合は、次のようにして成果物の品質計測を実行し、これによって指定されたタイミング毎に品質計測を

繰り返して実行する(202)。

【0031】すなわち、まず、ターミナル品質計測制御部22は、計測対象ターゲットファイルを参照し、該当する成果物の品質計測を行うように、ターミナル品質計測部21の起動制御を行う。

【0032】例えば、図4に示した計測タイミング指定ファイルの例では、92年9月18日に設計工程の成果物の品質計測を行うように設定されているため、ターミナル品質計測制御部22は、この計測タイミング指定ファイルを参照し、92年9月18日の時点でターミナル品質計測部21の起動を行う。

【0033】起動を指示されたターミナル品質計測部21は、指定されたディレクトリに存在する対象成果物の品質計測を行う。この際、品質の計測は、定型書式設計書およびソースコードの複雑さ、保守性、理解し易さ等を定量的に評価するための品質メトリクスによって計測される。

【0034】品質メトリクスとしては、設計書レベルではモジュールの数、条件の数、ループの数、処理の数、引数の数等を利用し、また、ソースコードレベルでは同様にモジュールの数、条件の数、ループの数、ステップの数、コメントの数、引数の数等を利用する。

【0035】上述のようにして計測された結果は、各ターミナル装置2の記憶部24に計測結果ファイルとして一時保管される(203)。

【0036】また、ターミナル品質計測制御部22は、計測実施状況(計測完了、未実施、失敗、実施日等)を、記憶部24の計測状況記録ファイルに記録する。この場合、品質計測を実行したので、計測状況ファイルに計測完了フラグをたてる(204)。

【0037】また、ターミナル品質計測制御部22は、ターミナル品質モニタ部23の起動指示を行い、各ターミナル装置2毎に、計測結果を確認可能な形の品質状況シートの作成を行う。これにより、必要に応じて各ターミナル装置2において、画面あるいは帳票によって計測結果を確認することができる(205)。

【0038】ホスト装置1では、各ターミナル装置2の計測完了フラグを確認することにより、システム制御部13が各ターミナル装置2の品質計測状況を常時監視する(102)。

【0039】そして、ホスト装置1で設定された計測タイミング指定ファイルの情報と対比して、指定されたタイミングで各ターミナル装置2の計測状況記録ファイルの計測実施状況を参照し、関連する全部のターミナル装置2の品質計測が完了したことを確認した上で(103)、各ターミナル装置2に一時保管されている計測結果ファイルを、ネットワークを介してホスト装置1に転送させる(104)。

【0040】そして、吸い上げた各ターミナル装置2の品質計測結果を基に、対象システム全体の品質を評価す

るため、これらのデータの集計を行い(105)、対象システム全体の品質値の算出および評価を行い、システム品質評価データを作成する(106)。このシステム品質評価データは品質データベース部12に保管される。

【0041】次に、ホスト装置1では、この対象ソフトウェア全体に関する品質データを基に、管理者に分かる形でその時点の品質状況の提示を行う(107)。品質状況の提示は、例えば図6および図7に示すように、帳票およびグラフ形式で行う。この出力は、計測終了後、プリンタによって自動的にプリントアウトされるが、操作画面からの設定入力を行うことにより、画面上にも表示することができる。ソフトウェア全体の品質状況の提示項目としては、以下の項目に関するものである。

【0042】 対象システム全体の品質評価値

対象システムを構成する各サブシステム(各ターミナル)の品質評価値

複数回の計測結果に基づく対象システムまたはサブシステムの品質評価値の変動傾向

図6に示した例は、あるシステム(システム名:\*\*\*\*)に関する品質評価結果および品質の動向を示したものである。本システムでは、計測した品質メトリクスデータ(例えば、モジュールの数、条件の数、ループの数、処理の数、等)を基に品質データ集計部11で、複雑さ、理解性、保守性等の観点からデータを集計し、最終的には、複雑さ、理解性、保守性それぞれについて0~10の間の点数を算出する。

【0043】例えば、複雑さについて殆ど問題がない場合には、「複雑さ=0」逆にたいへんに複雑で問題があると考えられる場合には、「複雑さ=10」となり、複雑さの点数は、この0から10の間の値をとる。図6に示した例は、「複雑さ=4」、「理解性=6」、「保守性=4」程度の場合の評価例である。

【0044】また、図7に示した帳票の例は、本システムによって定期的に品質計測を実施した結果を基に、品質評価値の変動状況を示したものである。この例では、あるシステム(システム名:\*\*\*\*)の保守性は、92年8月10日の計測では3であったものが、92年8月20日の計測では4になり、92年8月30日の計測では再度3に低下し、92年9月10日の計測では5となっていること等を示している。

【0045】各ターミナル装置2から吸い上げられた品質計測結果および品質データ集計部11で集計加工されたシステム品質評価データは、品質データベース部12に保管される。保管された品質データは、品質評価情報を提供する際の過去の参考値として利用される。なお、本システムの品質評価では、過去の値を基に、各品質メトリクス毎の評価基準値を設定し、これを利用する。

【0046】以上説明したように、本実施例のソフトウェア品質評価システムによれば、次のような効果を得る

ことができる。

【0047】すなわち、分散環境下でのソフトウェア開発における全体の品質評価を自動的に行うことが可能となる。また、品質計測のタイミングを予め設定して品質計測・評価が実行できるため、開発途中段階での品質確認並びに品質作り込みが容易に行える。さらに、品質データの計測・収集に関する作業が自動化されることにより、品質計測・評価に要する工数が削減され、ソフトウェアの生産性向上に寄与できる。

【0048】また、品質計測タイミングを短期スパンで設定することにより、品質状況の変化を確認できるため、品質劣化等の傾向を早期に発見することが可能となり、より効果的な対策が可能となる。また、過去のソフトウェア開発に関する品質計測結果を基にした評価基準値を利用することにより、より精度の高い評価が可能となる。

【0049】なお、本システムは、分散環境下でのソフトウェア開発における品質データ計測・評価を行うものであるが、品質データ集計部11、品質状況提示部14等を利用して、工程管理データ等の集計・提示を行うことも可能である。すなわち、本システムでは、計測タイミング指定ファイルで指定されたタイミングで定期的に個人の作業成果物の品質計測を行うので、この際に作業成果物のサイズ（ステップ数）等を計ることにより、作業の進捗データとしての利用も可能となり、これによって作業進捗状況の定量的評価を行うことができる。

【図3】

```

Terminal:EOS
Design Measurement Target      :Yesq¥tffdesign

Source Code Measurement Target  :Yesq¥source
  
```

【図7】

システム品質動向 92-8-10				
システム名: ****				
月 日	92-8-10	92-8-20	92-8-30	92-9-10
複雑さ	5	5	4	5
理解性	4	5	5	5
保守性	3	4	3	5

## 【0050】

【発明の効果】以上説明したように、本発明のソフトウェア品質評価システムによれば、分散環境下でソフトウェア開発を行う場合であっても、ソフトウェア全体の品質評価を容易に行うことができる。

## 【図面の簡単な説明】

【図1】本発明の一実施例のソフトウェア品質評価システムの構成を示す図。

【図2】品質評価を行う対象成果物の例を示す図。

【図3】計測対象ターゲットファイルの例を示す図。

【図4】計測タイミング指定ファイルの例を示す図。

【図5】図1のシステムの動作を説明するための図。

【図6】品質評価結果の提示例を示す図。

【図7】品質動向の評価結果の提示例を示す図。

## 【符号の説明】

- 1 ホスト装置
- 2 ターミナル装置
- 11 品質データ集計部
- 12 品質データベース
- 13 システム制御部
- 14 品質状況提示部
- 20 品質状況モニタ
- 21 ターミナル品質計測部
- 22 ターミナル品質計測制御部
- 23 ターミナル品質モニタ部
- 24 記憶部

【図6】

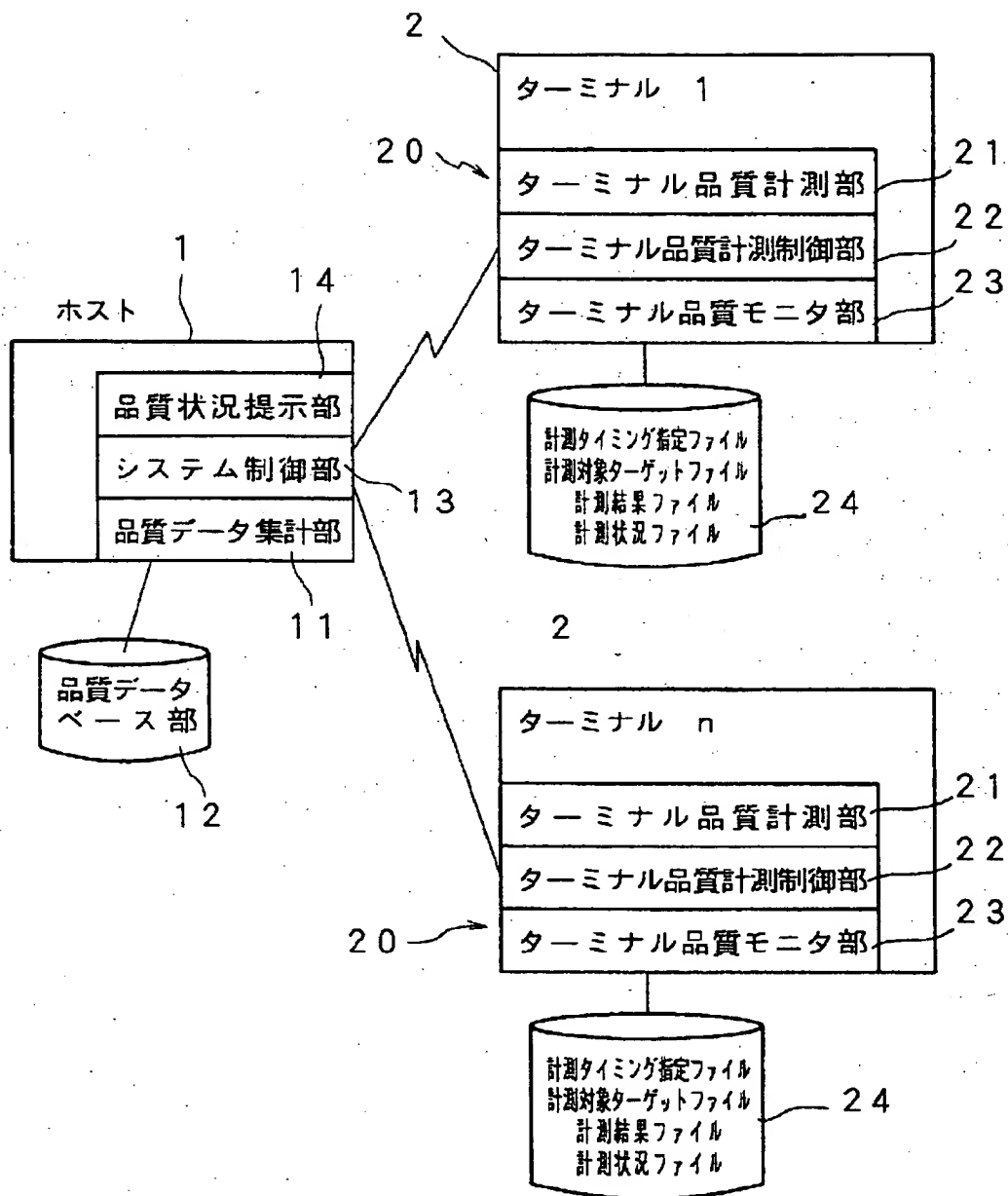
```

システム品質評価結果 92-8-10

システム名: ****

複雑さ  0  ▽  10
理解性  0  ▽  10
保守性  0  ▽  10
  
```

【図1】



**定形形式設計書**

**TFF 設計書**

```

graph TD
    Start([start]) --> Step1[処理の初期設定をする]
    Step1 --> Step2[同じディレクトリのファイル名を取]
    Step2 --> Step3[計測ターゲットファイルからデ  
1行づつ読み込む]
    Step3 --> Step4[読み込みデータを...]
  
```

**フローチャート**

```

graph TD
    In(( )) --> Step1[命令受信]
    Step1 --> Decision{命令コード受信完了?}
    Decision -- Yes --> Out1(( ))
    Decision -- No --> Loop(( ))
    Loop --> Step1
  
```

**ソースコード**

**C言語**

```

1  #include "stdio.h"
2  #define N 1000
3  int main() {
4      char *buf = (char *)malloc(N);
5      if (buf == NULL) return -1;
6      while(1) {
7          printf("Enter string: ");
8          fgets(buf, N, stdin);
9          printf("%s\n", buf);
10         if(feof(stdin)) break;
11     }
12     free(buf);
13     return 0;
14 }
```

**COBOL**

```

1  IDENTIFICATION DIVISION.
2  PROGRAM-ID . TFF .
3  DATA DIVISION.
4  FILE SECTION.
5  FD INFILE.
6  CONTAINS INFILE.DAT.
7  BEGINNING OF FILE.
8  OPEN INFILE.
9  PERFORM GETLINE.
10 END PROGRAM.
```

**品質評価結果**

本装置 → 品質評価結果



【図4】

Measurement Target: Sample.trg  
Measurement Schedule

92-8-01	Design-1
92-8-15	Design-2
92-9-18	Design-3
92-9-30	Source-1
92-10-30	Source-2

【図5】

